Land Information System (LIS) Test Plan
Submitted under Task Agreement GSFC-CT-2
Cooperative Agreement Notice (CAN)
CAN-00OES-01
Increasing Interoperability and Performance of Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences

March 17, 2003

Version 1.1

CONTENTS

1.0 Scope

This Test Plan describes the software test plan for the Land Information System (LIS). LIS is a project to build a high-resolution, high-performance land surface modeling and data assimilation system to support a wide range of land surface research activities and applications.

This document has been prepared in accordance with the requirements of Task Agreement GSFC-CT-2 under Cooperative Agreement Notice CAN-00-OES-01 Increasing Interoperability and Performance of Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences, funded by NASA's ESTO Computational Technologies Project.

1.1 Land Information System

The Land Information System will be a near real-time, high-resolution (up to 1 km) global land data simulation executed on highly parallel computing platforms, with well defined, standard-conforming interfaces and data structures to interface and interoperate with other Earth system models, and with flexible and friendly web-based user interfaces.

1.2 Land Surface Modeling and Data Assimilation

In general, land surface modeling seeks to predict the terrestrial water, energy and biogeochemical processes by solving the governing equations of the soil-vegetation-snowpack medium. Land surface data assimilation seeks to synthesize data and land surface models to improve our ability to predict and understand these processes. The ability to predict terrestrial water, energy and biogeochemical processes is critical for applications in weather and climate prediction, agricultural forecasting, water resources management, hazard mitigation and mobility assessment.

In order to predict water, energy and biogeochemical processes using (typically 1-D vertical) partial differential equations, land surface models require three types of inputs: 1) initial conditions, which describe the initial state of land surface; 2) boundary conditions, which describe both the upper (atmospheric) fluxes or states also known as "forcings" and the lower (soil) fluxes or states; and 3) parameters, which are a function of soil, vegetation, topography, etc., and are used to solve the governing equations.

The proposed LIS framework will include various components that facilitate global land surface modeling within a data assimilation system framework. The main software components of the system are:

- LDAS (Land Data Assimilation System) : is a software system that integrates the use of land surface models in a data assimilation framework.
- Land surface Models : LIS will include 3 different land surface models, namely, CLM, NOAH, and VIC.

These components are explained in detail in the following sections.

1.3 Land Data Assimilation System (LDAS)

LDAS is a model control and input/output system (consisting of a number of subroutines, modules written in Fortran 90 source code) that drives multiple offline one dimensional land surface models (LSMs) using a vegetation defined "tile" or "patch" approach to simulate sub-grid scale variability. The one-dimensional LSMs such as CLM and NOAH, which are subroutines of LDAS, apply the governing equations of the physical processes of the soil-vegetation-snowpack medium. These land surface models aim to characterize the transfer of mass, energy, and momentum between a vegetated surface and the atmosphere.

LDAS makes use of various satellite and ground based observation systems within a land data assimilation framework to produce optimal output fields of land surface states and fluxes. The LSM predictions are greatly improved through the use of a data assimilation environment such as the one provided by LDAS. In addition to being forced with real time output from numerical prediction models and satellite and radar precipitation measurements, LDAS derives model parameters from existing topography, vegetation and soil coverage's. The model results are aggregated to various temporal and spatial scales, e.g., 3 hourly, 0.25 deg x 0.25 deg.

The execution of LDAS starts with reading in the user specifications. The user selects the model domain and spatial resolution, the duration and timestep of the run, the land surface model, the type of forcing from a list of model and observation based data sources, the number of "tiles" per grid square (described below), the soil parameterization scheme, reading and writing of restart files, output specifications, and the functioning of several other enhancements including elevation correction and data assimilation.

The system then reads the vegetation information and assigns subgrid tiles on which to run the one-dimensional simulations. LDAS runs its 1-D land models on vegetation-based "tiles" to simulate variability below the scale of the model grid squares. A tile is not tied to a specific location within the grid square. Each tile represents the area covered by a given vegetation type.

Memory is dynamically allocated to the global variables, many of which exist within Fortran 90 modules. The model parameters are read and computed next. The time loop begins and forcing data is read, time/space interpolation is computed and modified as necessary. Forcing data is used to specify boundary conditions to the land surface model. The LSMs in LDAS are driven by atmospheric forcing data such as precipitation, radiation, wind speed, temperature, humidity, etc., from various sources. LDAS applies spatial interpolation to convert forcing data to the appropriate resolution required by the model. Since the forcing data is read in at certain regular intervals, LDAS also temporally interpolates time average or instantaneous data to that needed by the model at the current timestep. The selected model is run for a vector of "tiles", intermediate information is stored in modular arrays, and output and restart files are written at the specified output interval.

1.4 Community Land Model (CLM)

CLM (Community Land Model) is a 1-D land surface model, written in Fortran 90, developed by a grass-roots collaboration of scientists who have an interest in making a general land model available for public use. LDAS currently uses CLM version 2.0, formerly known as the Common Land Model. CLM version 2.0 was released in May 2002 and is implemented in the current version of LDAS/LIS. The source code for CLM 2.0 is freely available from the National Center for Atmospheric Research (NCAR) (http://www.cgd.ucar.edu/tss/clm/). The CLM is used as the land model for the Community Climate System Model (CCSM) (http://www.ccsm.ucar.edu/), which includes the Community Atmosphere Model (CAM) (http://www.cgd.ucar.edu/cms/). CLM is executed with all forcing, parameters, dimensioning, output routines, and coupling performed by an external driver of the user's design (in this case done by LDAS). CLM requires pre-processed data such as the land surface type, soil and vegetation parameters, model initialization, and atmospheric boundary conditions as input. The model applies finite-difference spatial discretization methods and a fully implicit time-integration scheme to numerically integrate the governing equations. The model subroutines apply the governing equations of the physical processes of the soil-vegetation-snowpack medium, including the surface energy balance equation, Richards' (1931) equation for soil hydraulics, the diffusion equation for soil heat transfer, the energy-mass balance equation for the snowpack, and the Collatz et al. (1991) formulation for the conductance of canopy transpiration.

1.5 The Community NOAH Land Surface Model

The community NOAH Land Surface Model is a stand-alone, uncoupled, 1-D column model freely available at the National Centers for Environmental Prediction (NCEP; ftp://ftp.ncep.noaa.gov/pub/gcp/ldas/noahlsm/). The name is an acronym representing the various developers of the model (N: NCEP; O: Oregon State University, Dept. of Atmospheric Sciences; A: Air Force (both AFWA and AFRL - formerly AFGL, PL); and H: Hydrologic Research Lab - NWS (now Office of Hydrologic Dev -- OHD)). NOAH can be executed in either coupled or uncoupled mode. It has been coupled with the operational NCEP mesoscale Eta model (Chen et al., 1997) and its companion Eta Data Assimilation System (EDAS) (Rogers et al., 1996), and the NCEP global Medium-Range Forecast model (MRF) and its companion Global Data Assimilation System (GDAS). When NOAH is executed in uncoupled mode, near-surface atmospheric forcing data (e.g., precipitation, radiation, wind speed, temperature, humidity) is required as input. NOAH simulates soil moisture (both liquid and frozen), soil temperature, skin temperature, snowpack depth, snowpack water equivalent, canopy water content, and the energy flux and water flux terms of the surface energy balance and surface water balance. The model applies finite-difference spatial discretization methods and a Crank-Nicholson time-integration scheme to numerically integrate the governing equations of the physical processes of the soil vegetation-snowpack medium, including the surface energy balance equation, Richards' (1931) equation for soil hydraulics, the diffusion equation for soil heat transfer, the energy-mass balance equation for the snowpack, and the Jarvis (1976) equation for the conductance of canopy transpiration.

1.6 Variable Infiltration Capacity (VIC) Model

Variable Infiltration Capacity (VIC) model is a macroscale hydrologic model, written in C, being developed at the University of Washington and Princeton University. The VIC code repository along with the model description and source code documentation is publicly available at http://hydrology.princeton.edu/research/lis/index.html. VIC is used in macroscopic land use models such as SEA - BASINS (http://boto.ocean.washington.edu/seasia/intro.htm). VIC is a semi-distributed, grid-based hydrological model, which parameterizes the dominant hydrometeorological processes taking place at the land surface - atmospheric interface. The execution of VIC model requires preprocessed data such as precipitation, temperature, meteorological forcing, soil and vegetation parameters, etc. as input. The model uses three soil layers and one vegetation layer with energy and moisture fluxes exchanged between the layers. The VIC model represents surface and subsurface hydrologic processes on a spatially distributed (grid cell) basis. Partitioning grid cell areas to different vegetation classes can approximate sub-grid scale variation in vegetation characteristics. VIC models the processes governing the flux and storage of water and heat in each cell-sized system of vegetation and soil structure. The water balance portion of VIC is based on three concepts:

1) Division of grid-cell into fraction sub-grid vegetation coverage.
2) The variable infiltration curve for rainfall/runoff partitioning at the land surface.
3) A baseflow/deep soil moisture curve for lateral baseflow.

Water balance calculations are preformed at three soil layers and within a vegetation canopy. An energy balance is calculated at the land surface. A full description of algorithms in VIC can be found in the references listed at the VIC website.

1.7 System Overview

The software system to be tested for the first code improvement is a functioning LDAS and two land surface models namely CLM, and NOAH. The system has been modified from serial code to parallel code and the resolution will be increased to 5km for the first code improvement.

The software system to be tested for the interoperability prototype is a functioning LDAS and three land surface models namely CLM, NOAH, and VIC. The system will be run on the SGI Origin and the LIS Cluster.

The software system to be tested for the second code improvement is a functioning LDAS and three land surface models namely CLM, NOAH, and VIC. The resolution will be increased to 1 km for the second code improvement. The web-based user interface and the data management system will be added. The system will run on the LIS Cluster. The testing needed for the second code improvement is not included in this version of the test plan.

The software system to be tested for subsequent milestones is TBD and is not included in this initial version of the test plan.

2.0 Referenced Documents

The test plan refers to the following documents:

- SGI Origin 3000 (http://www.nas.nasa.gov/user/systemsdoc/O3K/o3k.html)
- LIS Requirements Document on the LIS web site (http://lis.gsfc.nasa.gov/documentation/)
- LIS Software Design Document on the LIS web site (http://lis.gsfc.nasa.gov/documentation/)
- CLM Land Surface Model. (http://www.cgd.ucar.edu/tss/clm)
- NOAH Land Surface Model. (ftp://ftp.ncep.noaa.gov/pub/gcp/ldas/noahlsm/)
- VIC Land Surface Model. (http://hydrology.princeton.edu/research/lis/index.html)

3.0 Software Test Environment

3.1 Hardware Platform

The LIS software tests will take place on the SGI Origin 3000 located at NASA Ames Research Center or the custom designed LIS Beowulf cluster. Refer to the documentation available from http://www.nas.nasa.gov/user/systemsdoc/O3K/o3k.html. for information on the SGI Origin 3000. The LIS Linux cluster is pictured below.

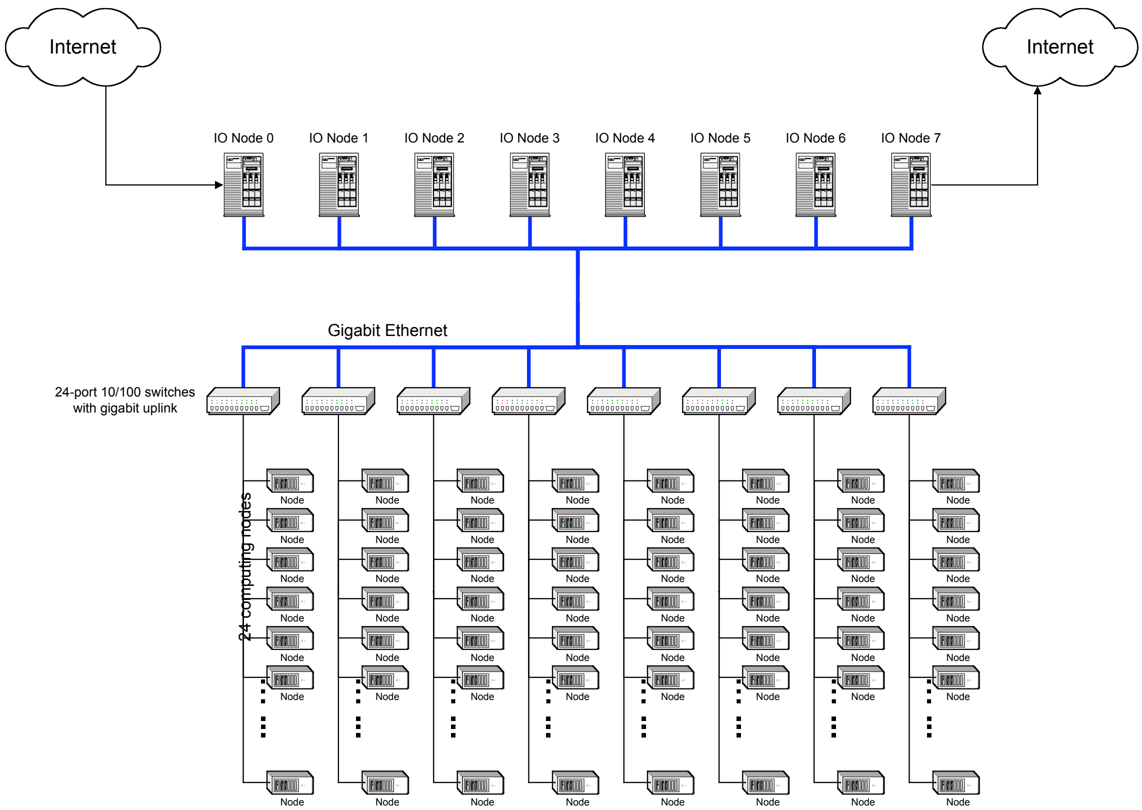

Figure 1: The physical architecture of the LIS Linux cluster. The cluster has 8 IO nodes and 192 compute nodes. Each IO node has dual Athlon CPUs, 2GB RAM and Gigabit NICs, and each compute node has a single Athlon CPU , 512MB RAM and a Fast Ethernet NIC.

3.2 Software Build being Tested

The software for the first code improvement will be in CVS under LIS_MILESTONE_F_TESTING. The software for the Interoperability Prototype will be in CVS under LIS_MILESTONE_I_TESTING.

3.3 Drivers and Test Tools

The drivers and test tools needed for these tests are customized PBS batch queuing scripts, SpeedShop performance tools, and Vampir performance tools.

3.4 Compilers and Libraries

The following compilers are used: SGI's f90, SGI's cc, Absoft's f90, and GNU's gcc.
The libraries needed for testing on the SGI are:

Message Passing Interface (SGI's MPI, MPICH)
Hierarchical Data Format
w3lib for GRIB libraries

bacio for low level byte wise I/O routines
Ipolates (iplib) for interpolation
gfio for DAO I/O library

3.5 Input Data Sets

The input data sets will be available from CVS. The input data sets will be forcing or parameter data called for by the land surface models. It is categorized as GEOS data, AGRMET data, NRL data, GVEG data, or BCS data.

3.6 Configuration of the test environment

- Checkout appropriate build from CVS repository
- Edit namelist parameters in ldas.crd file
- Edit the PBS batch queuing script

4.0 Test Identification

4.1 General Information

This initial test plan covers test in calendar year 2003. The test plan will be updated to include details and additional tests for calendar year 2004.

All tests are planned at the system level.

4.2 Planned Tests for First Code Improvement

This section describes testing planned for Milestone F due by March 2003.

4.2.1 5km Input data Validation
Purpose: Validate newly generated 5km input data sets.
Type/class:
Test Inputs: Manually created 5km data
Verification methods: PI review
Special requirements: na
Assumptions/Constraints: na
Expected results: PI judges data acceptable
Actual results:
Discrepancy reports:

4.2.2 5km CLM on SGI
Purpose: Verify that CLM V2.0 will run on the SGI Origin at 5km resolution
Type/class: Acceptance

Test Inputs: 5km input data sets, card file
Verification methods: PI review
Special requirements: na
Assumptions/Constraints: na
Expected results: PI judges output data acceptable
Actual results:
Discrepancy reports:

4.2.3 5km NOAH Run on SGI Origin
Purpose: Verify that NOAH V2.5 will run on the SGI Origin at 5km resolution
Type/class: acceptance
Test Inputs: 5km input data sets, card file
Verification methods: PI review
Special requirements: na
Assumptions/Constraints: na
Expected results: PI judges output data acceptable
Actual results:
Discrepancy reports:

4.2.4 5km CLM run on SGI Origin
Purpose: Verify that CLM V2.0 will run on the SGI Origin at a 5km resolution at a throughput of 1ms per grid cell per day.
Type/class: performance
Test Inputs: 5km input data sets, card file
Verification methods: Throughput value will be calculated from run-time information.
Special requirements: na
Assumptions/Constraints: na
Expected results: Throughput of 1 ms per grid cell per day
Actual results:
Discrepancy reports:

4.2.5 5km NOAH run on SGI Origin
Purpose: Verify that NOAH V2.5 will run on the SGI Origin at a 5km resolution at a throughput of 1ms per grid cell per day.
Type/class: performance
Test Inputs: 5km input data sets, card file
Verification methods: Throughput value will be calculated from run-time information.
Special requirements: na
Assumptions/Constraints: na
Expected results: Throughput of 1 ms per grid cell per day.
Actual results:
Discrepancy reports:

4.3 Planned Test for Interoperability Prototype

This section describes testing planned for Milestone I due by July 2003.

4.3.1 Interoperability Test
Purpose: Demonstrate code interoperability of VIC land surface model on SGI and LIS cluster.
Type/class: acceptance
Test Inputs: input data sets, card file
Verification methods: successful completion of 1 day run and comparison of outputs.
Special requirements: na
Assumptions/Constraints: na
Expected results: Successful completion of 1 day run on both platforms and outputs are reviewed.
Actual results:
Discrepancy reports:

4.3.2 5km NOAH Run on LIS Cluster
Purpose: Verify that NOAH V2.5 will run on the LIS Cluster at 5km resolution
Type/class: acceptance
Test Inputs: 5km input data sets, card file
Verification methods: PI review
Special requirements: na
Assumptions/Constraints:na
Expected results: PI judges output data acceptable
Actual results:
Discrepancy reports:

4.3.3 5km CLM run on LIS Cluster
Purpose: Verify that CLM V2.0 will run on the LIS Cluster at a 5km resolution..
Type/class: acceptance
Test Inputs: 5km input data sets, card file
Verification methods: PI judges output data acceptable
Special requirements: na
Assumptions/Constraints: na
Expected results: Throughput of 1 ms per grid cell per day
Actual results:
Discrepancy reports:

4.4 Planned Tests for Second Code Improvement

This section will describe testing planned for Milestone G due by February 2004. Listed below is a partial list of tests that will be updated in mid-2003.

4.4.1 Verify 1km Input data
Type/class: acceptance

4.4.2 1km CLM Run on LIS Cluster
Type/class: acceptance

4.4.3 1 km NOAH Run on LIS Cluster
Type/class: acceptance

4.4.4 1km VIC Run on LIS Cluster
Type/class: acceptance

4.4.5 1km NOAH Run on Linux Cluster
Type/class: performance

4.4.6 1km CLM Run on Linux Cluster
Type/class: performance

4.4.7 1km VIC Run on Linux Cluster
Type/class: performance

4.4.8 TBD

4.5 Planned Tests for Interoperability Demonstration

This section describes testing planned for Milestone J due by July 2004.
TBD

4.6 Planned Tests for Customer Delivery

This section describes testing planned for Milestone K due in August 2004.
TBD

5.0 Test Schedule

Tests for section 4.2	March 2003
Tests for section 4.3	July 2003
Tests for section 4.4	TBD
Tests for section 4.5	TBD
Tests for section 4.6	TBD

6.0 Requirements Traceability

The current version of the LIS Requirements Traceability Matrix is available from http://lis.gsfc.nasa.gov/documentation/CT/.